Exploring Interpretability for Machine Learning

Mahak Agarwal
Viterbi School of Engineering, University of Southern California, Los Angeles, California 90089, USA

(Dated: May 1, 2021)

Neural Networks can undoubtedly achieve great performance on a majority of tasks nowadays. But along with that, there is a growing need for them to be able to explain their decisions. They are famously called 'black boxes', which means it is extremely difficult to understand their outputs and how they reached there. This paper aims to extend the works in [1] and explore different interpretability techniques and understand their effectiveness, for the task of object detection. This paper has used convolutional neural networks but these techniques can be tested for different tasks and networks as well. This paper explored visualizing channel attributions, features, and activations of neuron groups to understand more about what the network is learning. It was observed that these techniques in combination with interactivity can make neural networks more explainable in particular cases, not all. For a better understanding in this particular task, with Inception V1, visualizing neurons with the highest activations and also neuron groups gives more information in understanding what the neural network understands at each layer. For VGG-19, visualizing neuron groups was the most effective.

Keywords: Machine Learning, Interpretability, Neural Networks, Convnets

I. INTRODUCTION

Convolutional neural networks contain convolutional layers, pooling layers and fully connected layers. Convolutional layers extract features from the image using filters and make feature maps, while the pooling layers help in reducing the dimensionality of the feature maps.[2] Finally, the fully connected layers, in the case of classifying an image, takes the results of the pooling layers and applies it, to output the classification label.
Even though we theoretically can explain the working and architecture of a convnet, it is still unclear how the model transitions the results from one layer to another. The main goal of interpretability is to keep information "human-scale" [1], in a sense, therefore, taking inspiration from [1], this paper tries to explore different techniques, and summarized below which techniques are effective in different cases, and which are not.

The task chosen here is of object detection, though as a future work, these techniques can be tested for another task as well. The model used is Inception V1 or GoogleNet, since it has been claimed in [1] to have neurons which are unusually meaningful semantically. In the later sections, the paper also explores other models like VGG-19, to see how effective these techniques are in their case.

II. HIDDEN LAYERS IN NEURAL NETWORKS

Understanding hidden layers is one of the most challenging areas of neural networks. Previous research has focused on understanding the input and output layers of neural networks, where in reality it's important to make sense of the hidden layers to make a robust and reliable model.

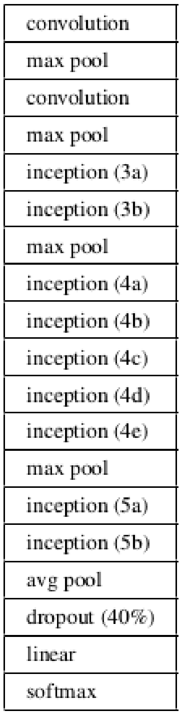

FIG. 1. Architecture of InceptionV1 model [3]

GoogleNet is 27 layers deep (Figure 1) and trained on the ImageNet dataset. It contains a number of convolution, pooling and inception layers(combinations of different sized convoluational layers) followed by a softmax layer in the end.

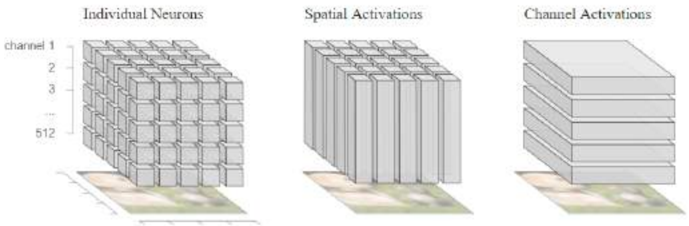

FIG. 2. Structure of individual neurons, spatial activations, channel activations in hidden layers of a network [1]

A. Attribution techniques

The first technique tested was channel attribution[1] for the inception layer mixed4d, since at this layer in GoogleNet, objects are detected by the network.[1]The layers before that concentrate on other low-level parts of the image. For example, layer conv2d0 detects edges, layer3a detects textures and layer4a detects patterns.

Therefore, we can visualise other layers, but meaningful and human understandable results for this task(object detection) will be visible in the inception layer mixed4d. One notable observation about ImageNet that affected the results for GoogleNet for this task, is the fact that the dataset seems to have the most number of images for various breeds of dogs. Even though it roughly has the same number of images for each category, since, there a large number of dog breeds represented in the dataset, therefore, total number of dog images become large. Therefore, the feature visualizations of channels seem to give a good result for an image containing any arbitrary breed of dog.

For example, if an image of a German shepherd is used as an input (Figure 3) and the activations and feature maps are compared, for the case if class is ”German shepherd” versus ”Maltese dog”, we can see that the feature maps are able to detect different features of the dog in the image and have more confidence that the image contains a German shepherd instead of a Maltese dog.

This is an impressive result since it is visible to the bare eye that the network is able to detect parts like sharp ears, snout and eyes which make it confident that the image contains a German shepherd.

But, if the same technique is applied to other non-frequent categories like ”cheeseburger” and ”hotdog” (Figure 4), the results are not very clear as to what the network sees. This shows that this technique works well with some categories but not all of them and even

[1] to see what part of input image causes the channel of the layer to behave in a particular way [4]

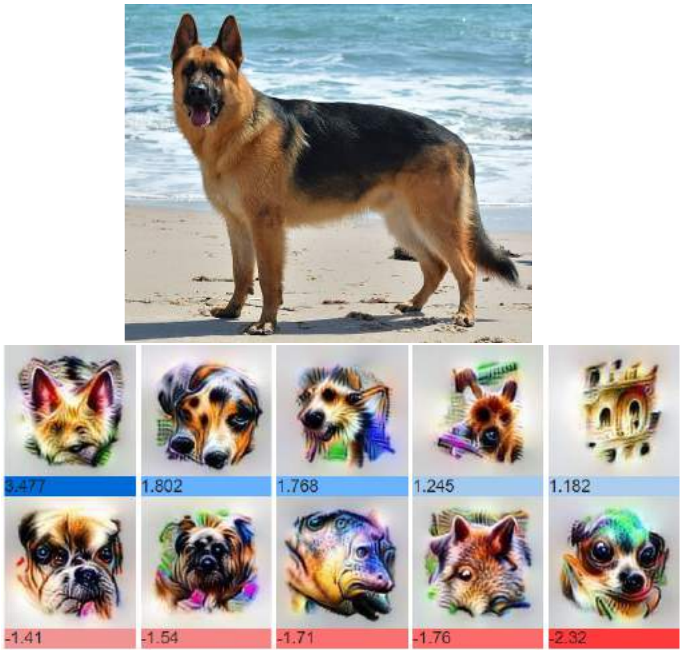

FIG. 3. **(Top)** An image of a German shepherd **(Bottom)** The first row contains feature visualizations of five neurons that most support the class ”German shepherd”. The second row contains feature visualizations of five neurons that that most support ”Maltese dog” along with their activation value

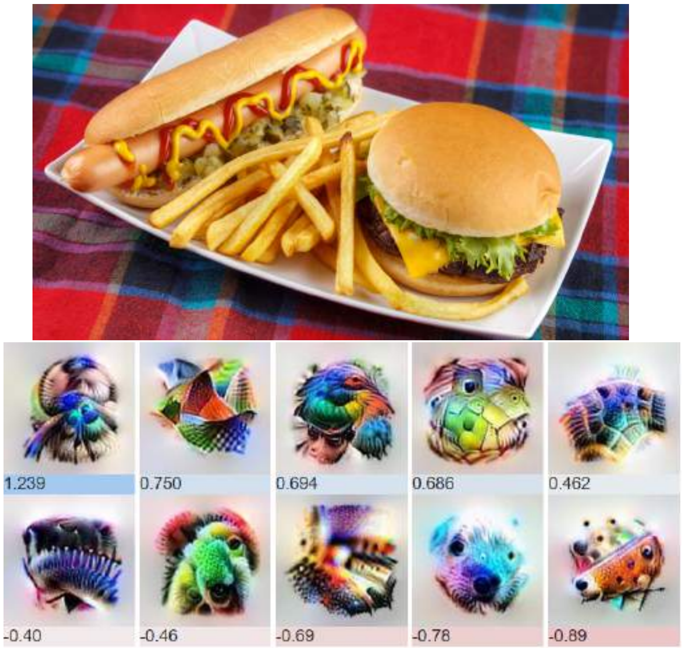

FIG. 4. **(Top)** An image of a Cheeseburger and Hotdog **(Bottom)** The first row contains feature visualizations of five neurons that most support the class ”Cheeseburger”. The second row contains feature visualizations of five neurons that that most support ”hotdog” along with their activation value

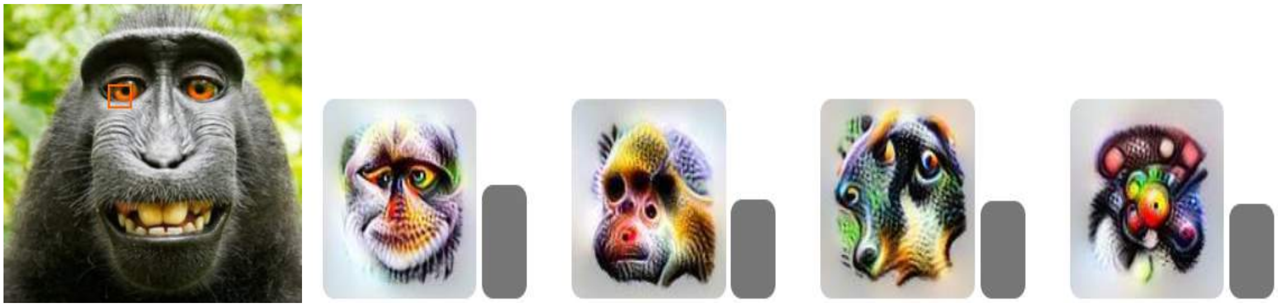

FIG. 5. **(Left)** An image of a monkey face **(Right)** Feature visualizations of the neurons with the highest activations (in descending order) for the highlighted part of the image

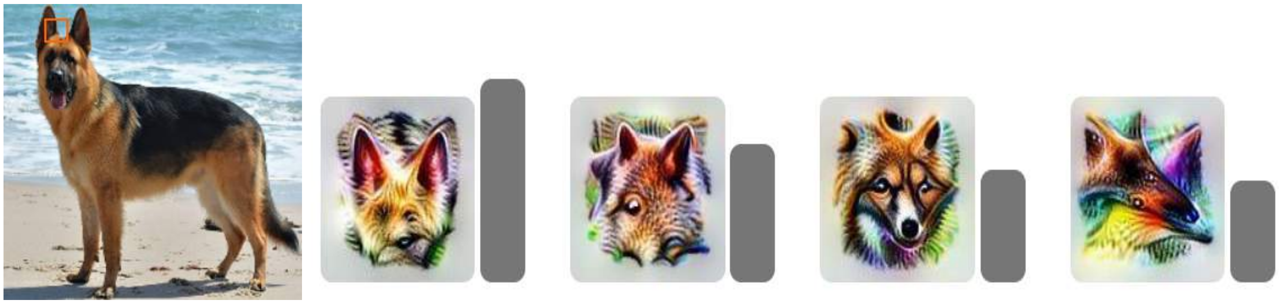

FIG. 6. **(Left)** An image of a German Shepherd dog **(Right)** Feature visualizations of the neurons with the highest activations (in descending order) for the highlighted part of the image

if the model is going in the right direction of detecting the cheeseburger and hotdog, the feature visualizations of channels do not give a clear picture to be able to be explanable to humans.

B. Network's interpretation of the input

In one sense, the neural network learns a new representation of the image at each layer.[1] Therefore, it might be useful if we can visualise at each layer, the neuron which gets the most fired up for a particular part of the image. This can be called as sort of a ”semantic dictionary” [1] as now the vectors containing activations have a visual representation. Earlier it was difficult to understand what these vectors meant since they contained an array of numbers(activations), but now with an image representation associated with each neuron activation, it is easier to comprehend the hidden layers.

Adding interactivity to this technique, by allowing the user to hover over the image and choose what part of the image they want to see the visualizations for, this technique gives a much broader understanding to the user of what is happening inside the neural network in real time. This technique was tried on different images (Figure 5 and 6). In Figure 5, we can see that when the chunk of area around the eye is given as input to the layer mixed4d, it fires the neurons which are able to recognize the eye area and general facial structure around the eyes. Similarly, if we try this technique for the ears of a German shepherd (Figure 6), we see that the neurons which have the highest activation, are the ones that contain shapes similar to ears in their feature maps.

C. Transforming for easy perception

One more objective of interpretability is to avoid overwhelming the user with huge amounts of information. Even after having meaningful visualizations for individual neurons and channels, it can be too tedious for a user to look at each visualization to understand the model. Therefore, the last technique explored in this paper is to combine spatial and channel activations to form neuron groups in a way that some aspect of those neuron groups are prioritized. There are different ways in which these neuron groups can be formed, to focus on gradient, attributions or activations. This paper forms neuron groups by prioritizing activations. For example,in Figure 7, the image of pineapple and banana is effectively decomposed into neuron groups which are the most fired up by particular areas of the image. To make it easier to understand, the areas are color-coded. Therefore, bananas -in ”red” fire up a particular group of neurons, which are visualized below it. Even though the feature visualization of neuron group may not be very clear to a human, information about which part of the image affect that neuron group, gives important information about the model overall about what it is able to understand.

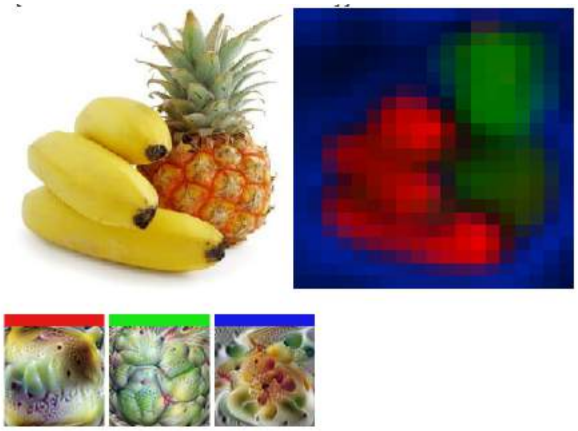

FIG. 7. **(Top Left)** An image of pineapple and banana.**(Top Right)** Color-coded activations of neuron groups.**(Bottom)** Visualizations of the different color-coded neuron groups.

III. MODIFICATION FOR THE CURRENT SCENARIO

This section explores interpretability techniques for the VGG-19 model. Main objective behind this section is to compare how VGG-19 performs with the above techniques for the task of object detection as compared to the Inception V1 model used in [1].

This paper first explores visualizing neuron groups and compares the output with the results obtained from the Inception V1 model. As highlighted by Figure 8 and Figure 9, VGG-19 seems to perform better. The layer used for VGG-19 is conv5_3/conv5_3, which is a one of the last layers in the architecture. The reason for using this layer is that higher level layers capture objects in the image while lower layers capture edges, textures etc. The layer used for Inception V1 is 'mixed4d', since it has been claimed and used in [1] and said to capture objects. For the image of Labrador retriever and tiger cat, it is able to detect the floppy ears, eyes and snout of the dog but not the cat's face, which should have been detected as well. Whereas, for the case of VGG-19, it is able to detect the dog's features as well as the cat face.

Further, visualizing neurons of VGG-19, validated the claims of [1] that Inception V1 has unusally semantic neurons. The neurons for VGG-19 were not very meaningful, due to which the first two techniques mentioned in this paper did not yield very good results. The spritemap for VGG-19 layer conv5_3/conv5_3 and Inception V1 layer mixed4d is attached in Appendix A for reference and comparison.

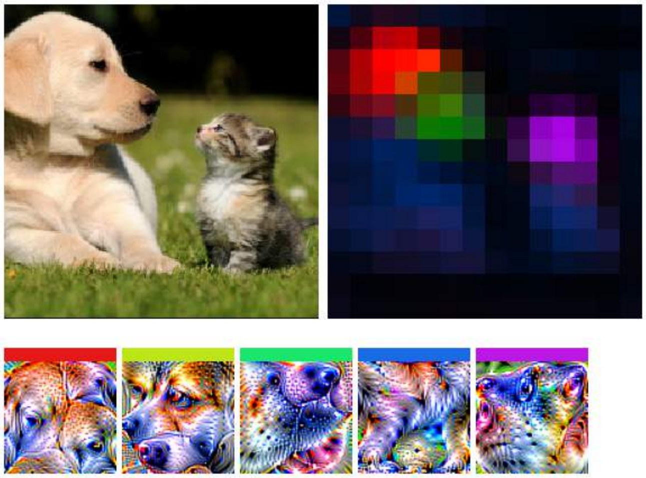

FIG. 8. **(Top Left)** An image of Labrador Retreiver and tiger cat.**(TopRight)** Color-coded activations of neuron groups with VGG-19.**(Bottom)** Visualizations of the different color-coded neuron groups

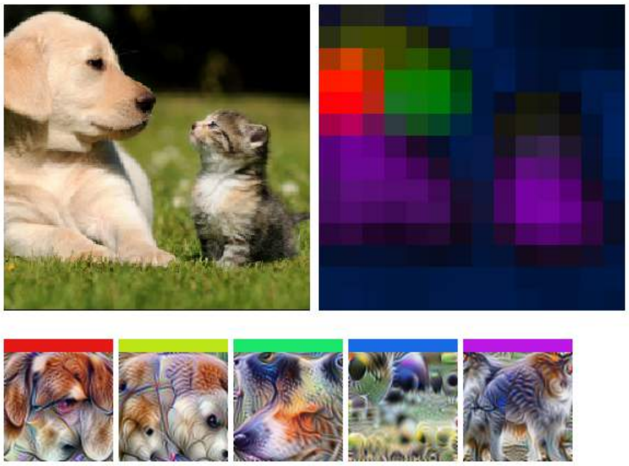

FIG. 9. **(Top Left)** An image of Labrador Retreiver and tiger cat.**(TopRight)** Color-coded activations of neuron groups with Inception V1.**(Bottom)** Visualizations of the different color-coded neuron groups

IV. DISCUSSION AND FUTURE WORK

After observing various images with the above techniques, it is clear that neural networks might be 'grey boxes' sometimes, but definitely not 'black boxes'. They can be understood and interpreted. Interactive visualizations can make understanding hidden layers easier. These inferences can be helpful for some applications where deep learning models have to be made more interpretable, either because they need to showcased to a huge population or because they will be used for crucical decision making. But the work in this paper needs to extended to explore these techniques for tasks other than object detection to see if they perform equally well there too.

As a future work, some or all of the techniques can be tested for other deep learning tasks with different models. Moreover, the main challenge lies in bringing interpretability to text or speech where the results are not automatically understandable.

V. CONCLUSIONS

Overall, the techniques highlighted above have a huge potential when it comes to understanding neural networks. However, if we compare different techniques, then visualizing neurons in a layer with the highest activations and forming a semantic dictionary seems to be more helpful in general while channel attribution with feature visualization can help only in some cases. Other than these, visualizing neuron groups is also helpful since it gives important information about what the network understands in a human friendly way. We also see that these results were true for Inception V1 and and if we apply the above techniques to another model like VGG-19 which is also trained on ImageNet, then visualizing neuron groups gives a better result compared to Inception V1. But, it does not have semantically meaningful neurons due to which the first two techniques do not give good results for it.

DATA AVAILABILITY

Data is available here

CODE AVAILABILITY

Code is available at Github
Colab notebook 1
Colab notebook 2
Colab notebook 3

ACKNOWLEDGMENTS

I would like to thank my Professor Dr.Marcin Abram, the Teaching Assistant Ninareh Mehrabi, for helping at various stages during the project and Manikanta Chunduru Balaji for giving constructive feedback on the draft. I would also like to thank my parents for supporting me throughout the task.

[1] C. Olah, *The building blocks of interpretability*, Distill 3.3 **e10** (2018).
[2] C. Szegedy, *Going deeper with convolutions*, Proceedings of the IEEE conference on computer vision and pattern recognition. (2015).
[3] A. Vidhya, *Understanding Inception Network from scratch*, .
[4] C. Olah, *Feature visualization*, Distill 2.11 **e7** (2017).

Appendix A: Additional Content

We can compare the spritemaps generated for a particular layer of the model, to see how meaningful the neurons are. It contains neuron visualizations of that layer. As we can see from Figure 10 and 11, Inception V1 layer mixed4d has more semantically meaningful neurons as compared to VGG-19 layer conv5_3/conv5_3.

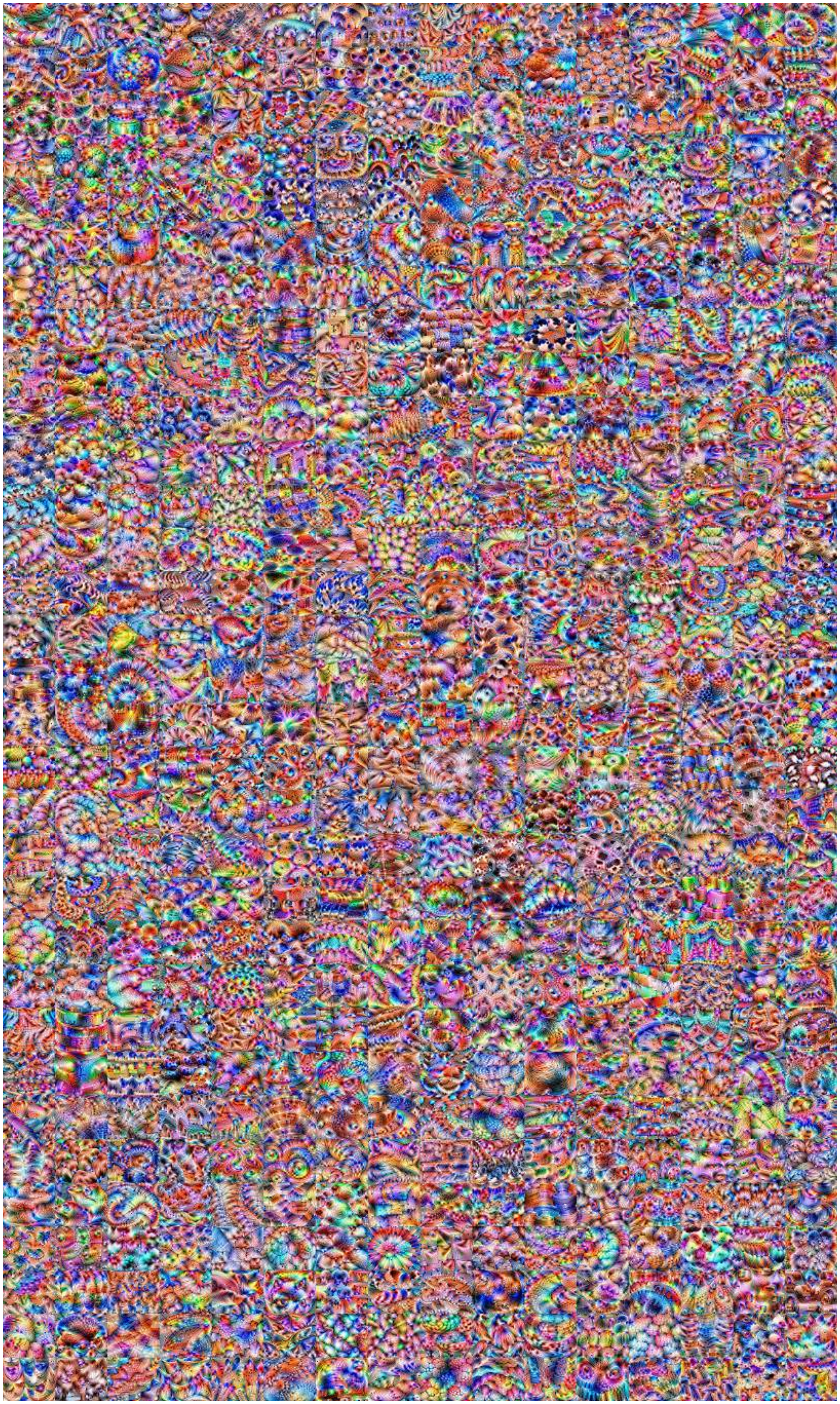

FIG. 10. Spritemap for layer conv5_3/conv5_3 of VGG-19. The spritemap above contains neuron visualizations of the 512 neurons in this layer.

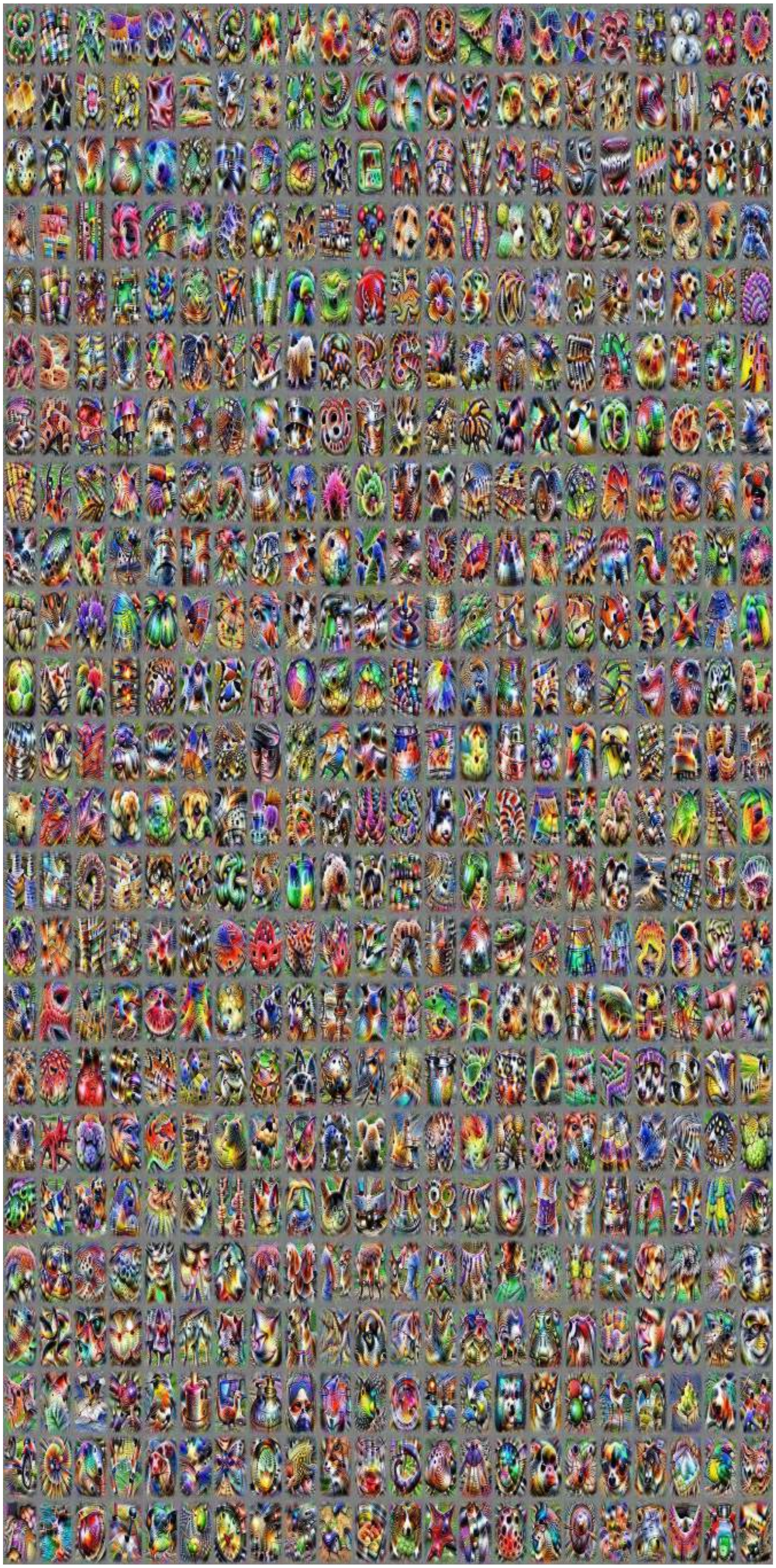

FIG. 11. Spritemap for layer mixed 4d of Inception V1. The spritemap contains neuron visualizations of the 528 neurons in this layer.[1]